[**Cơ sở trí tuệ nhân tạo - CQ2021/1**](https://courses.fit.hcmus.edu.vn/course/view.php?id=3898)



**BÁO CÁO ĐỒ ÁN**

Tên đồ án: **AI\_Lab1 – Các thuật toán tìm kiếm**

Nhóm: **21**

MSSV 1: 21120127 Họ tên: Lê Hoàng Sơn

Giáo viên lý thuyết: Bùi Duy Đăng

Giáo viên lý thuyết: Lê Nguyễn Nhựt Trường

C:\Users\tdqua_000\Dropbox\SS-Slides\DeCuong-CDIO\Template CDIO v4.2\Templates\Hinh anh\LogoTruong.png

Bộ môn Công nghệ phần mềm

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

# C:\Users\tdqua_000\Desktop\self_reflection.jpg TỰ ĐÁNH GIÁ ĐỒ ÁN

|  |  |  |  |
| --- | --- | --- | --- |
| **Mức** | **Mô tả** | **Tỷ lệ hoàn thành** | **Phần trăm điểm** |
| Mức 1 (40%) | cài đặt thành công 1 thuật toán để tìm đường đi từ S tới G. Báo  cáo lại thuật toán và quá trình chạy thử. Lưu ý, chạy thử trường hợp không có  đường đi. | 100% | 40% |
| Mức 2 (30%) | cài đặt ít nhất 3 thuật toán khác nhau (ví dụ tìm kiếm mù, tham  lam, heuristic, …). Báo cáo nhận xét sự khác nhau khi chạy thử 3 thuật toán. | 100% | 30% |
| Mức 3 (20%) | trên bản đồ sẽ xuất hiện thêm một số điểm khác được gọi là  điểm đón. Xuất phát từ S, sau đó đi đón tất cả các điểm này rồi đến trạng thái  G. Thứ tự các điểm đón không quan trọng. Mục tiêu là tìm ra cách để tổng  đường đi là nhỏ nhất. Báo cáo thuật toán đã áp dụng và quá trình chạy thử | 100% | 20% |
| Mức 4 (10%) | các hình đa giác có thể di động được với tốc độ h tọa độ/s.  Cách thức di động có thể ở mức đơn giản nhất là tới lui một khoảng nhỏ để  đảm bảo không đè lên đa giác khác. Chạy ít nhất 1 thuật toán trên đó. Quay  video và đính kèm trực tiếp/link vào báo cáo. | 100% | 10% |
| Điểm cộng (10%) | thể hiện mô hình trên không gian 3 chiều (3D) | 100% | 10% |

**Điểm tự đánh giá cho toàn đồ án**: 110% / **100%**.

## Mô tả dự án

**a. Tên của dự án**: “Gizmo”

**b. Git:** <https://github.com/lhson0606/Path-Finder-py.git>

**c. Môi trường thực thi**: Windows 11

**d. Tools:** RenderDocs, Github Copilot, Chat GPT, aserprite, PyCharm. Github.

## Đóng góp của các thành viên cho dự án

**Tỉ lệ đóng góp**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ và tên** | **Tỉ lệ** |
| 01 | 21120127 | Lê Hoàng Sơn | 100% |
|  |  | **Tổng:** | 100% |

**Chi tiết các công việc đã thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **SV thực hiện** | **Tên chức năng / công việc** | **Chú ý** |
| 01 | Lê Hoàng Sơn | All | Một thành viên |

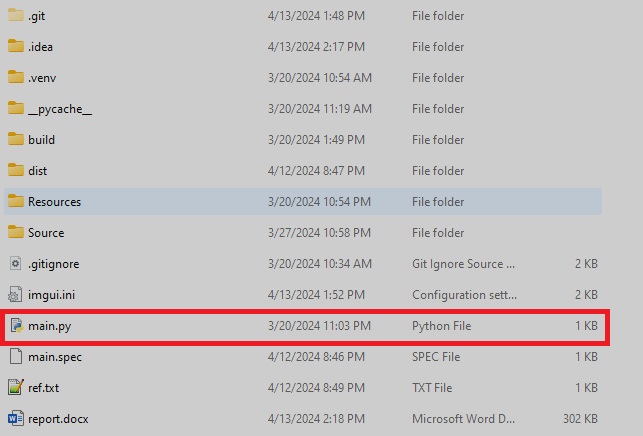
## Thông tin cần thiết để thực thi chương trình

* *Cài đặt môi trường Python: 3.10.8 hoặc mới hơn.*
* *Các package cần thiết:*
  + *PyGLM (vec3, matrix, …)*
  + *PyOpenGL (graphic)*
  + *imgui (debug GUI)*
  + *numpy (multi-dimension array)*
  + *glfw (manage window)*
  + *esper (light-weight ECS framework)*
  + *colorLog (colorful console logging)*
  + *tk (file dialog)*
  + *pillow (image support)*

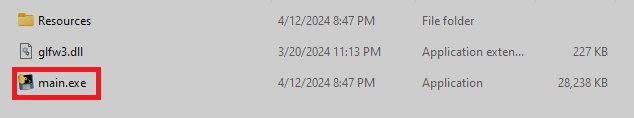
*Để install các package:* ***pip install PyGLM PyOpenGL imgui numpy glfw esper colorLog tk pillow***

*Install từng package:*

* + ***pip install PyGLM***
  + ***pip install PyOpenGL***
  + ***pip install imgui***
  + ***pip install numpy***
  + ***pip install glfw***
  + ***pip install esper***
  + ***pip install colorLog***
  + ***pip install tk***
  + ***pip install pillow***
* *Sau khi đã cài đặt môi trường vào thư mục Project “Gizmo”, chạy command “****py main.py****” hoặc double-click vào “****main.py****” để chạy*



***Note:*** *Có thể chạy file thực thi “main.exe” được build sẵn trong folder* ***dist***

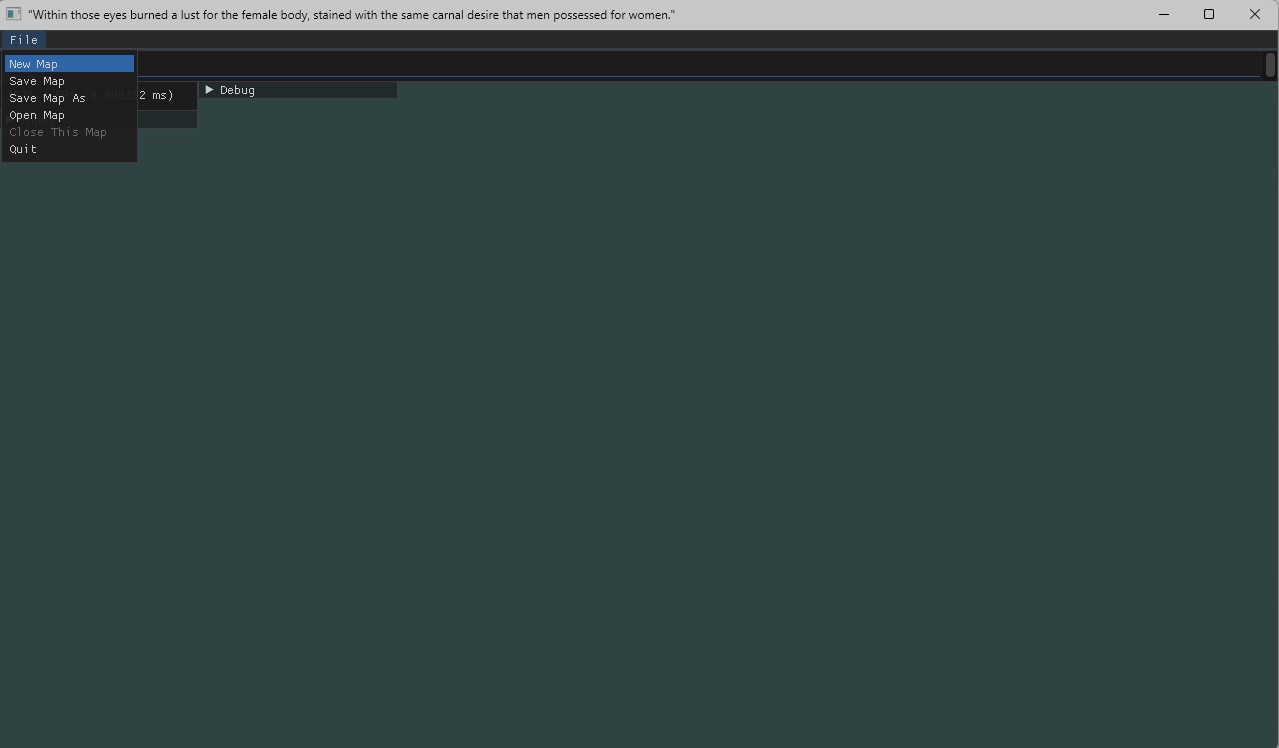
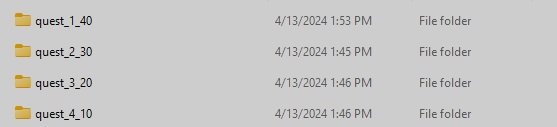
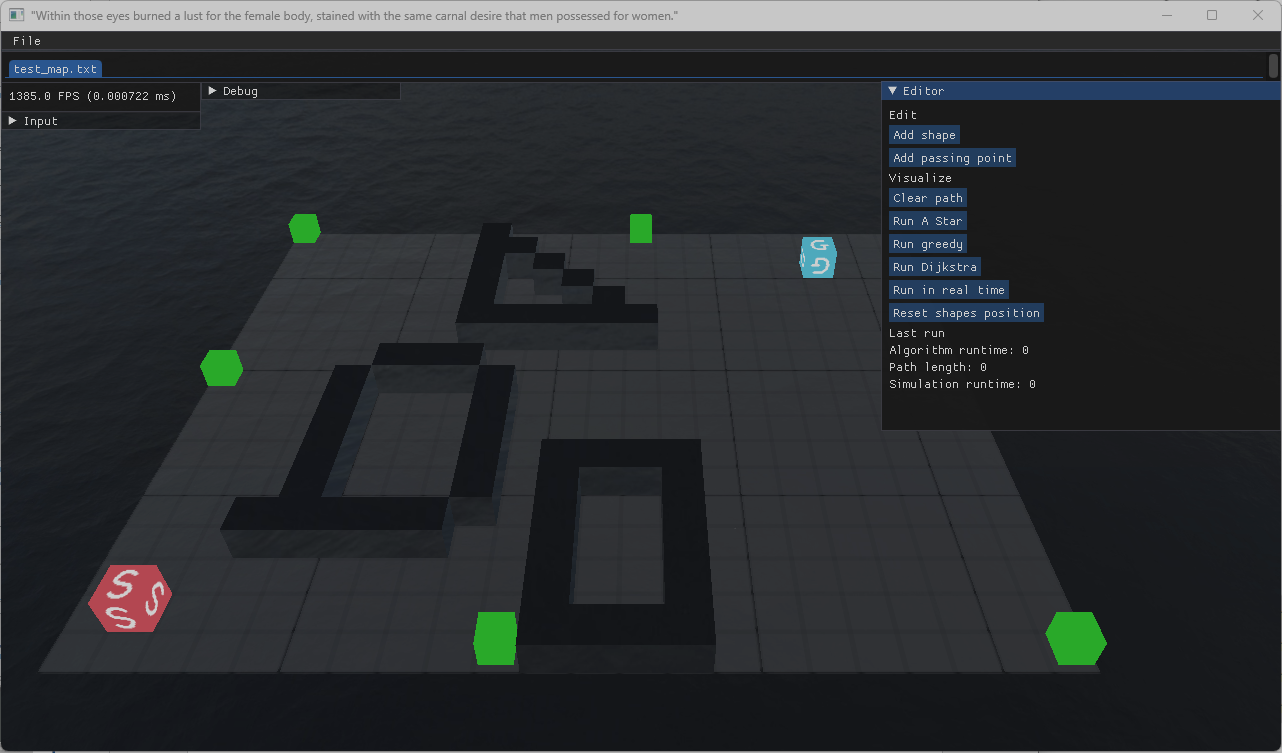
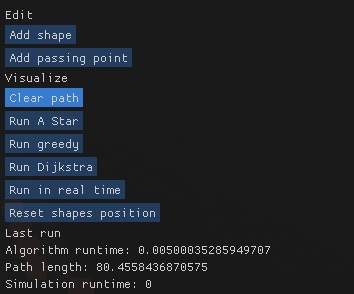
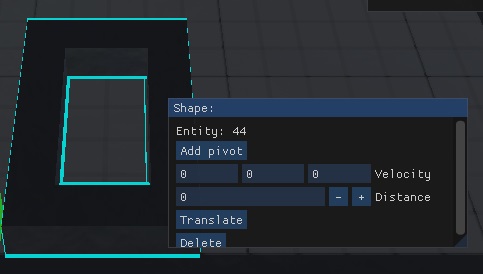
******

***Phần cứng khuyến nghị:***

*Intel - 10500K or equivalent/higher*

*RTX 2080 or equivalent /higher*

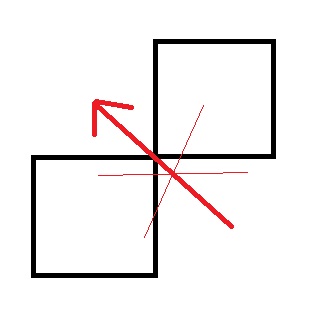
## User guidelines:

* Người dùng vào Menu “**File**” chọn “**New map**“để tạo một map mới (tính năng chưa hoàn thiện), hoặc chọn **“Open map”** để mở map tạo sẵn. Có 4 folder chứa map của 4 mức, mỗi folder chứa 03 map để test.
* 
* 
* 
* Điều khiển: ấn giữ chuột phải, sau đó dùng các phím W,A,S,D để di chuyển. Lên xuống bằng Q/E và Shilft/Spacebar. Để thay đổi góc nhìn, giữ chuột phải và di chuyển chuột.
* 
* Chức năng **“Add shape”** và **“Add passing point”** tạm thời chưa được hoàn thành.
* Để test các thuật toán lần lượt dùng **“Run A Star”**, **“Run greedy”**, **“Run Dijkstra”** hoặc để chạy khi các shape di chuyern chọn **“Run in real time”.**
* Để thao tạo với shape (ví dụ thêm tốc độ, xóa, thêm điểm mốc, …) nhập chuột trái vào shape cần chọn.
* 
* Để thoát edit một shape ấn phím **“ECS”**
* 

*(Ấn ESC để thoát)*

# CÁC MỨC ĐÃ THỰC HIỆN

**Lưu ý: Để hạn chế việc đi xuyên qua đa giác, nhóm đã sử dùng thuật toán kiểm tra có đi vào đa giác không. Tư tưởng kiểm tra xem khi đi chéo thì hai ô kề có phải là tưởng hay không.**

****

Mã giả mà nhóm đã thực hiện:

**def** is\_moveable(from, to, map):

**if** (from.x == to.x):

off\_set = to - from

**if** (map.is\_wall(from.x, from.y+off\_set.x, from.z) **and** map.is\_wall(from.x, from.y, from.z + off\_set.y))

**return False**

//tương tự cho y và z

**return True**

## Mức 1: Cài đặt thành công 1 thuật toán

*Thuật toán đã cài đặt* ***A Star****:*

Pseudocode: [https://en.wikipedia.org/wiki/A\*\_search\_algorithm#Pseudocode](https://en.wikipedia.org/wiki/A*_search_algorithm%23Pseudocode)

Kết quả thu được khi chạy trong folder **“quest\_1\_40”**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Input** | **Image** | **Output** | **Profile** |
| map1.txt | 22,18 2,2,19,16 3 4,4,5,9,8,10,9,5 8,12,8,17,13,12 11,1,11,6,14,6,14,1 |  |  |  |
| map2.txt | 22,18,4 2,2,0,19,16,3 3 4,4,1,5,9,2,8,10,0,9,5,3 8,12,1,8,17,0,13,12,2 11,1,0,11,6,0,14,6,0,14,1,0 |  |  |  |
| map3.txt | 20,20 2,2,19,16 1 6,20,15,20,15,1,6,1 |  |  | Không có đường đi |

Báo cáo:

* Map 1: test map 2D, thuật toán chạy nhanh, có thể dùng trong thời gian thực (chỉ ~0.002 seconds). Search space vừa phải.
* Map 2: test map 3D với kích thước 22x18x4, thuật toán chạy bắt đầu chậm hơn hết ~0.1 sec. Vùng tìm kiếm lớn đáng kể, phải duyệt qua 1008 ô trước khi tìm được tới đích.
* Map 3: (map 2D) test trong trường hợp không tồn tại đường đi.

## Mức 2: cài đặt ít nhất 3 thuật toán khác nhau

03 thuật toán mà nhóm đã chọn là:

* A Star
* Greedy
* Dijkstra

Test map trong folder **“quest\_2\_30”**:

Kết quả chạy thử thuật toán **A Star**: *xem hình ở mục 2.1*

Kết quả chạy thử thuật toán **Greedy**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Input** | **Image** | **Output** | **Profile** |
| map1.txt | 22,18 2,2,19,16 3 4,4,5,9,8,10,9,5 8,12,8,17,13,12 11,1,11,6,14,6,14,1 |  |  |  |
| map2.txt | 22,18,4 2,2,0,19,16,3 3 4,4,1,5,9,2,8,10,0,9,5,3 8,12,1,8,17,0,13,12,2 11,1,0,11,6,0,14,6,0,14,1,0 |  |  |  |
| map3.txt | 20,20 2,2,19,16 1 6,20,15,20,15,1,6,1 |  |  | Không có đường đi |

Kết quả chạy thử thuật toán **Dijkstra**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Input** | **Image** | **Output** | **Profile** |
| map1.txt | 22,18 2,2,19,16 3 4,4,5,9,8,10,9,5 8,12,8,17,13,12 11,1,11,6,14,6,14,1 |  |  |  |
| map2.txt | 22,18,4 2,2,0,19,16,3 3 4,4,1,5,9,2,8,10,0,9,5,3 8,12,1,8,17,0,13,12,2 11,1,0,11,6,0,14,6,0,14,1,0 |  |  |  |
| map3.txt | 20,20 2,2,19,16 1 6,20,15,20,15,1,6,1 |  |  | Không có đường đi |

**Điểm chung:**

* Đường đi của các thuật toán có khác nhau đôi chút nhưng độ dài là như nhau ở các map.

**Khác biệt:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Time | Space | Search volume |
| A Star | Tùy input mà nhanh hay chậm hơn greedy. | Xấp xỉ Greedy | Nhỏ hơn so với Dijkstra |
| Greedy | Tùy input mà nhanh hay chậm hơn A Star. Ví dụ ở map 1 và 2 A Star lần lượt là 0.002 và 0.105 sec còn Greedy là 0.0075 và 0.06. | Xấp xỉ A Star | Tùy input nhưng thường nhỏ hơn so với A Star |
| Dijkstra | Chậm hơn nhiều ở tất cả các map. Ví dụ map 1 và 2 đều khá lâu: 0.0111 và 0.104 sec | Ít nhất, do không cần lưu trừ hàm f mà chỉ cần hàm g. | Lớn nhất so với hai thuật toán còn lại |

## Mức 3 thêm một số điểm là điểm đón.

Thuật toán mà nhóm cài đặt là **Dynamic Programming**

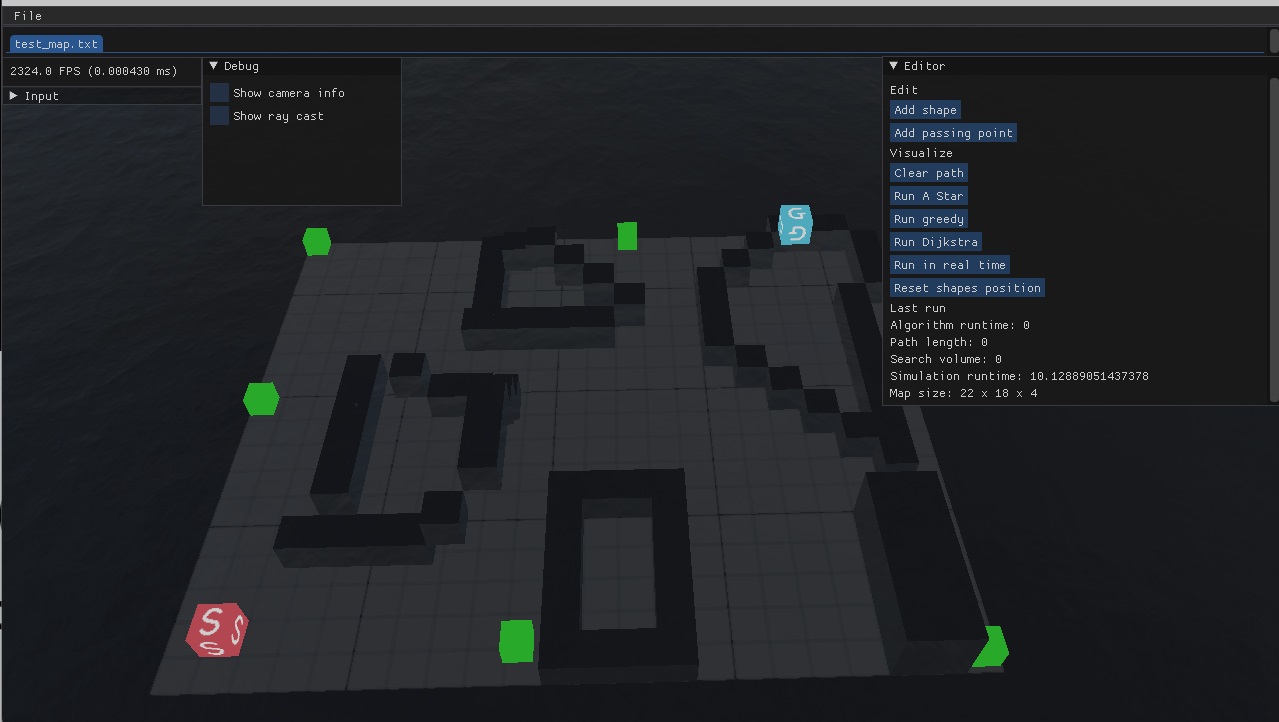
Test các map ở folder **“quest\_3\_20”**

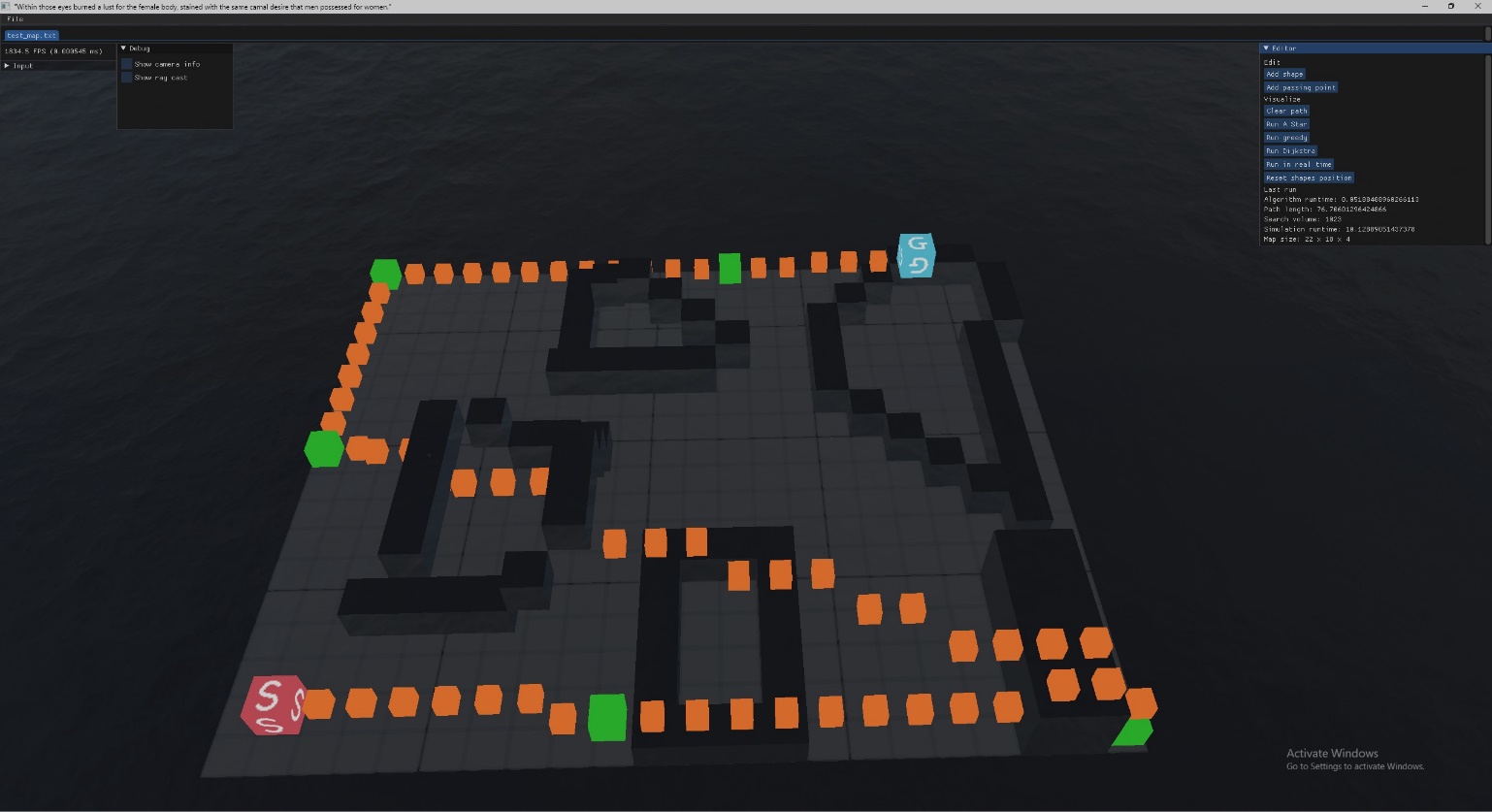
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Input** | **Image** | **Output** | **Profile** |
| map1.txt | 22,18 2,2,19,16,22,1,10,1,1,18,13,18,1,10 3 4,4,5,9,8,10,9,5 8,12,8,17,13,12 11,1,11,6,14,6,14,1 |  |  |  |
| map2.txt | 25,25,4 2,2,0,19,16,2,22,1,2,10,1,1,1,18,0,13,18,0,1,10,0,25,25,3 4 4,4,1,5,9,2,8,10,0,9,5,3 8,12,1,8,17,0,13,12,2 11,1,0,11,6,0,14,6,0,14,1,0 21,25,0,25,15,0,21,7,0,16,15,0 |  |  |  |
| map3.txt | 5,5,5 1,1,0,5,5,0,2,1,3,2,5,3 1 2,1,0,2,5,0,2,5,1,2,1,1,3,1,0,3,5,0,3,5,1,3,1,1 |  |  |  |

## Mức 4 các hình đa giác có thể di động được

Link demo: <https://youtu.be/9rcEufkZtpQ>

## 3D: Điểm cộng





# CÁC THAM KHẢO

**SÁCH**

<none>

**BÀI VIẾT / BÀI HƯỚNG DẪN**

1. **Learn OpenGL.**

Liên kết:

<https://learnopengl.com/>

2. **Mouse picking.**

Liên kết:

<https://www.youtube.com/watch?v=71G-PVpaVk8>,

<https://ogldev.org/www/tutorial29/tutorial29.html>

3. **A Star.**

Liên kết:

[https://en.wikipedia.org/wiki/A\*\_search\_algorithm](https://en.wikipedia.org/wiki/A*_search_algorithm)

4. **Greedy.**

Liên kết:

<none> (slide môn học)

5. **Dijkstra**:

Liên kết:

[https://en.wikipedia.org/wiki/Dijkstra%27s\_algorithm#:~:text=Dijkstra's%20algorithm%20(%2F%CB%88da%C9%AA,%2C%20for%20example%2C%20road%20networks.](https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm%23:~:text=Dijkstra's%20algorithm%20(%2F%CB%88da%C9%AA,%2C%20for%20example%2C%20road%20networks.)

6. **Object outlining.**

Liên kết:

<https://www.youtube.com/watch?v=ngF9LWWxhd0&t=328s>

7. **Grid texture**:

Liên kết:

<https://www.shadertoy.com/view/XtBfzz>